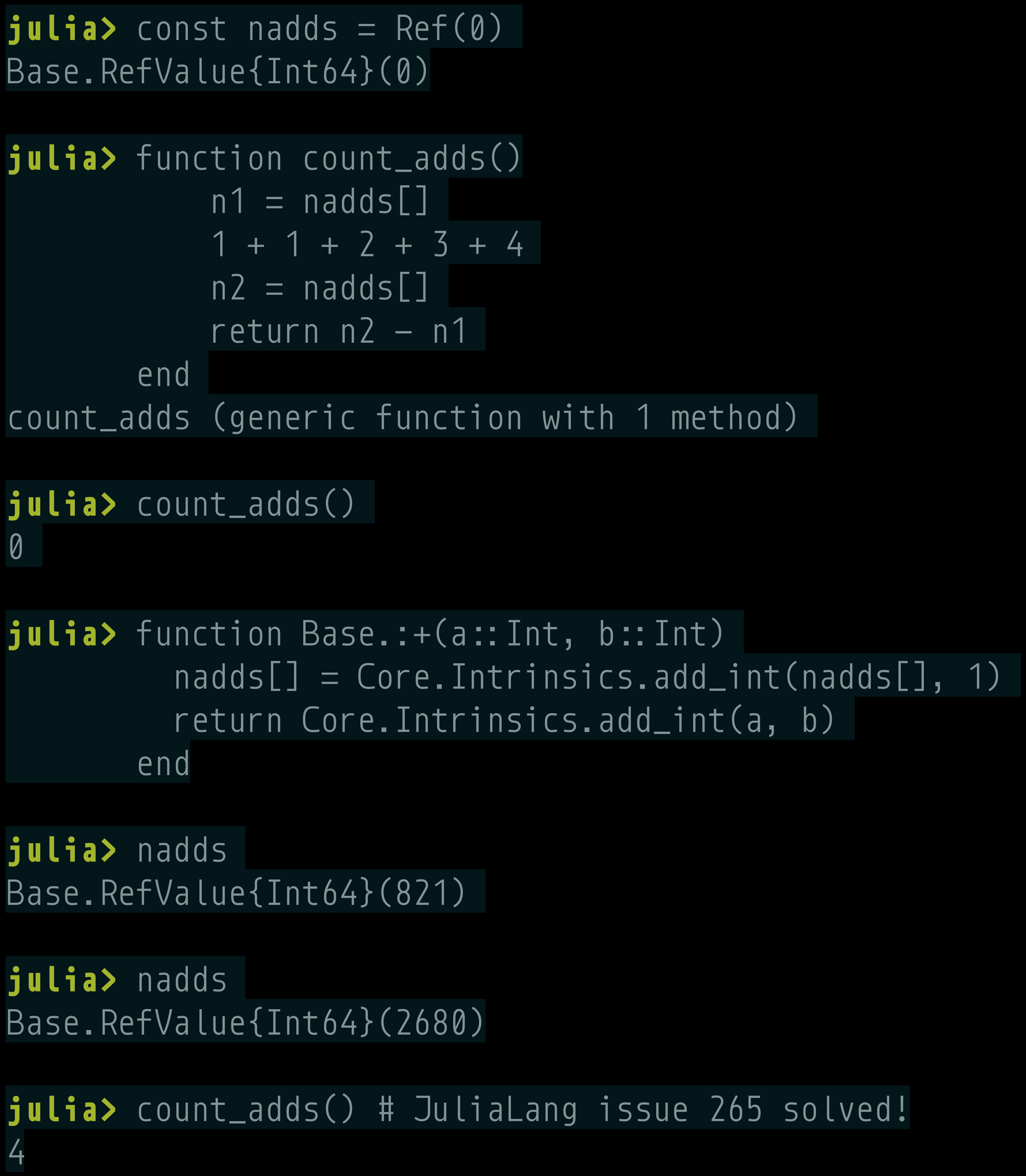

```
julia> const nadds = Ref(0)
Base.RefValue{Int64}(0)

julia> function count_adds()
           n1 = nadds[]
           1 + 1 + 2 + 3 + 4
           n2 = nadds[]
           return n2 - n1
       end
count_adds (generic function with 1 method)

julia> count_adds()
0

julia> function Base.:+(a::Int, b::Int)
         nadds[] = Core.Intrinsics.add_int(nadds[], 1)
         return Core.Intrinsics.add_int(a, b)
       end

julia> nadds
Base.RefValue{Int64}(821)

julia> nadds
Base.RefValue{Int64}(2680)

julia> count_adds() # JuliaLang issue 265 solved!
4
```

Method Backedges

- Adding a new method increments the world counter and starts invalidating intersecting method signatures
- Compute list of methods and old specializations that got replaced by this new method
- Remove from the method fast-dispatch caches
- Disable usage of old inference result in future worlds (truncate the max age)
- Recurse over all backedges